Dynamic action recognition based on Dynemes and Extreme Learning Machine

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki 54124, Greece Tel,Fax: +30-2310996304

{*aiosif,tefas,pitas*}*@aiia.csd.auth.gr*

Abstract

In this paper, we propose a novel method that performs dynamic action classification by exploiting the effectiveness of the Extreme Learning Machine (ELM) algorithm for single hidden layer feedforward neural networks training. It involves data grouping and ELM based data projection in multiple levels. Given a test action instance, a neural network is trained by using labeled action instances forming the groups that reside to the test sample's neighborhood. The action instances involved in this procedure are, subsequently, mapped to a new feature space, determined by the trained network outputs. This procedure is performed multiple times, which are determined by the test action instance at hand, until only a single class is retained. Experimental results denote the effectiveness of the dynamic classification approach, compared to the static one, as well as the effectiveness of the ELM in the proposed dynamic classification setting.

Keywords: Activity recognition; Dynamic classification; Fuzzy Vector Quantization; Extreme Learning Machine

1. Introduction

Human action recognition is a very active research field finding application in many important tasks, such as visual surveillance [1], human-computer interaction [2], augmented reality [3] and semantic video annotation [4]. Actions are usually described by using either features based on optical flow [5], or features devised mainly for action representation [6]. Although the use of such features leads to satisfactory action recognition results, their computation is expensive. Thus, when fast operation is important, action recognition methods should employ simpler action representations. Neurobiological studies [7] have concluded that the human brain can perceive actions by observing only the human body poses during action execution. Thus, actions can be described as sequences of consecutive human body poses, in terms of human body silhouettes [8, 9]. After describing actions, most methods in the literature exploit supervised machine learning techniques for action class representation and classification of new, unknown, action instances. Such techniques require a training phase, where labeled data are used in order to determine the system parameters. For example, in Artificial Neural Networks (ANNs) based data classification [10], training data are employed in order to de-

termine the neurons' weights and in Linear Discriminant Analysis (LDA) based data projection [11], labeled data are used in order to determine a mapping to a lower dimensional feature space for class representation and data classification. Traditionally, the training phase is performed offline by using the entire training set.

Action recognition is not an easy task, mainly due to the fact that there is not a formal description of actions. Variations between different action realizations resulted from different action execution styles and different human body sizes between persons result to high intra- and, possibly, low inter-action class variations. This is why person specific classification schemes have been recently investigated for action recognition [12]. The main idea in these classification schemes is to focus the classification problem on each individual person. That is, action recognition is performed by a classifier which has been trained by using action instances of the person under consideration. Following this approach, the above mentioned issues are effectively addressed leading to high action classification rates. However, the application of such classification schemes is limited, since, in order to operate properly, a person should have been recorded and trained before recognition. In different cases their performance will probably decrease.

An alternative choice could be the use of dynamic action classification schemes.

Dynamic classification, involves a system parameters adaptation procedure based either on the training set structure, or on the test data to be classified. Following this approach, several dynamic classification schemes have been proposed. Wright et. al. [13] proposed a dynamic classification scheme exploiting sparsity constraints. A given test sample is involved in a class independent regression procedure exploiting a codebook containing all the available labeled samples. Multiple reconstruction samples are, subsequently, produced by employing the labeled samples belonging to each class independently. Finally, the test sample is classified to the class providing the minimum reconstruction error. Tang et. al. [14] proposed the Dynamic Committee Machine (DCM), which employs five state-of-the-art classifiers (experts). A test sample is introduced to all the five classifiers and five classification results are produced. The dynamic nature of DCM is based on the adopted fusion strategy, where the experts' weights are modified depending on the corresponding test sample. Kyperountas et. al. proposed a dynamic classification scheme involving an iterative grouping procedure combined with LDA-based data classification [15]. The iterative procedure used in order to determine the optimal training set for LDA based data classification is intuitive and effective. However, the LDA based classification approach in this setting has two disadvantages: a) It sets the assumption of linear class separability. As it will be shown

in Section 5, this assumption is not met for action classes. b) The use of a small number of training data, compared to the training data dimensionality, leads to the small sample size problem [16]. In order to address this issue, Kyperountas et. al. employed an LDA variant proposed in [17], in which a regularization parameter should be a priori known and, thus, an offline training procedure is required. Finally, c) by using training data belonging to only two (or three) classes, LDA projection provides an one- (or two-) dimensional feature space, where classes discrimination may not be captured properly, especially for linear classification models.

In order to take into account the non linear nature of action classes, non linear classification methods should be employed. ANNs could be a good choice, since they have proven their effectiveness in a wide range of challenging classification problems. Among them, single hidden layer feedforward networks (SLFNs) have been widely used due to their ability to approximate any target continuous function and classify any disjoint regions. Furthermore, their operation is fast and, thus, they are appropriate for the cases where fast operation is important. However, most of the popular learning algorithms for SLFNs training are slow, due to their iterative nature, and their parameter values should be carefully chosen. This renders them inappropriate for dynamic classification schemes. Extreme

Learning Machine (ELM) [18] is a recently proposed algorithm for fast SLFNs training requiring much less human effort. By using a sufficiently large number of hidden neurons, the ELM classification scheme can be thought as a non linear mapping of the training data in a high dimensional feature space, noted as ELM space, followed by a linear classification procedure. Thus, non linear classification functions can been approximated. Furthermore, the ELM training procedure is independent of the training set size. These properties of ELM render it as a good choice for dynamic classification schemes.

In this paper we propose a novel dynamic classification method inspired from the above described dynamic subspace learning schemes and the effectiveness of the ELM training procedure. The proposed classification procedure can be seen as an adaptive multiple layer ANN, in which the number of layers, as well as the number of each layer neurons, are dynamically determined by the test action instance at hand, as illustrated in Figure 1. The proposed scheme is evaluated in action recognition by using the dyneme based action representation [19]. However, it can be easily modified in order to be employed for different action representations. It is efficient in the sense that it dynamically determines the optimal labeled data for training and classification. Furthermore, by exploiting the fast training procedure of the ELM, the classification procedure is fast and efficient.

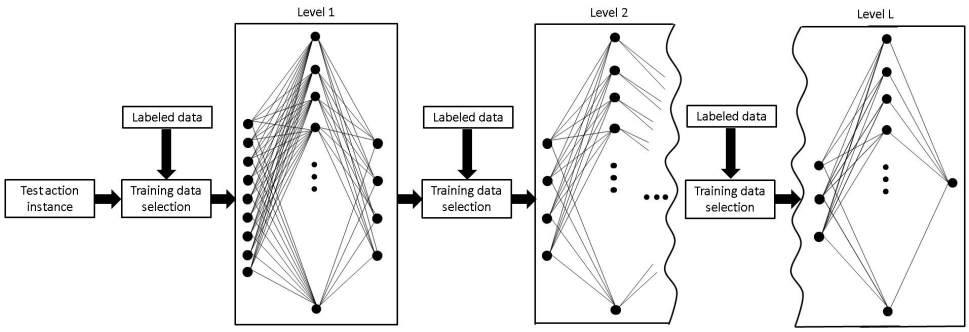


Figure 1: *Adaptive multiple layer network topology.*

The remainder of this paper is structured as follows. In Sections 2 we provide an overview of the adopted action representation. In Sections 3 and 4 we present the two calculation steps that will be used in Section 5 to describe the proposed dynamic classification method. Section 6 presents experiments conducted for assessing its performance. Finally, conclusions are drawn in Section 7.

2. Dyneme based action representation

In this section, we present an overview of the dyneme based action representation [19]. Let $\mathcal{A}$ be an action class set consisting of C action classes, such as walk, run, jump, drink, eat, etc. Let $\mathcal{U}$ denote an action recognition database containing N labeled action instances depicted in N videos, which will be called action videos hereafter. Video segmentation techniques, such as background subtraction [20] or color based image segmentation [21], are applied to the action

video frames in order to produce binary action videos depicting the human body poses. The video frames forming the binary action videos are centered to the human body regions of interest (ROIs), cropped to the ROIs region and resized to produce binary posture images of fixed ($H \times W$ pixels) size. In the experiments presented in this paper, we chose the size of the binary posture images to be equal to 32×32 pixels, which has been found experimentally to be a good compromise between computational cost and action recognition accuracy. The above described procedure is illustrated in Figure 2.

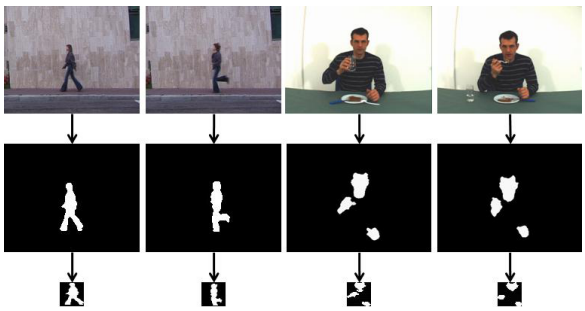


Figure 2: *Binary posture images production. From left to right 'walk', 'run', 'drink' and 'eat'.*

These binary images are represented as matrices, which are vectorized column-wise in order to produce the so called posture vectors $\mathbf{p}_{ij}$, $i = 1, ..., N$, $j = 1, ..., N_i$, where N_i denotes the number of binary images forming binary action video i. Posture vectors of all the N labeled binary action videos are clustered, without exploiting the available label information, in order to produce D action

independent representative posture vectors, the dynemes. This is done by applying D-Means clustering [11] to the posture vectors, minimizing the intra-cluster scatter, i.e.:

$$\sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{j=1}^{N_i}\alpha_{ijd}\|\mathbf{p}_{ij}-\mathbf{v}_d\|^2, \tag{1}$$

where $\alpha_{ijd}=1$, if $\mathbf{p}_{ij}$ is assigned to cluster d and $\alpha_{ijd}=0$, otherwise. Dynemes $\mathbf{v}_d,\ \ d=1,...,D$ are defined to be the cluster mean vectors, i.e.:

$$\mathbf{v}_d=\frac{1}{n_d}\sum_{i=1}^{N}\sum_{j=1}^{N_i}\alpha_{ijd}\mathbf{p}_{ij}. \tag{2}$$

After dynemes calculation, each posture vector $\mathbf{p}_{ij}$ is mapped to the membership vector $\mathbf{u}_{ij}\in\mathbb{R}^D$, which denotes the fuzzy similarity of $\mathbf{p}_{ij}$ with all the dynemes $\mathbf{v}_d$, according to a fuzzification parameter $m>1$:

$$u_{ijd}=\frac{(\parallel \mathbf{p}_{ij}-\mathbf{v}_d \parallel_2)^{-\frac{2}{m-1}}}{(\sum_{k=1}^{D}\parallel \mathbf{p}_{ij}-\mathbf{v}_k \parallel_2)^{-\frac{2}{m-1}}},\ \ d=1,...,D. \tag{3}$$

The optimal value of the fuzzification parameter m is obtained by applying the cross-validation procedure. Following [19], a value of $m=1.1$ has been used in all the experiments presented in this paper. Finally, action vectors $\mathbf{s}_i\in\mathbb{R}^D$ are calculated as the mean normalized membership vectors of the corresponding action videos:

$$\mathbf{s}_i=\frac{1}{N_i}\sum_{j=1}^{N_i}\mathbf{u}_{ij}. \tag{4}$$

Action vectors $\mathbf{s}_i$ representing all the training action videos are normalized in order to have zero mean and unit variance. Action vectors representing test action videos are normalized accordingly.

3. Data grouping and similarity measure

Let $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^{N_z}$ be a vector set consisting of N_z labeled vectors $\mathbf{z}_i$. In order to determine K vector groups in $\mathcal{Z}$, we apply a clustering technique without exploiting the available action class labels. Since K-Means is a fast clustering algorithm, we employ K-Means to this end. That is, $\mathcal{Z}$ is clustered by minimizing:

$$\sum_{k=1}^{K} \sum_{i=1}^{N_z} \beta_{ik} \|\mathbf{z}_i - \boldsymbol{\mu}_k\|^2. \tag{5}$$

$\boldsymbol{\mu}_k$ is the mean vector of group k, having cardinality $l_k = \sum_{i=1}^{l_k} \beta_{ik}$, i.e., $\boldsymbol{\mu}_k = \frac{1}{l_k} \sum_{i=1}^{l_k} \beta_{ik} \mathbf{z}_i$, and is used to represent the group. The number of groups K is either assumed to be known (fixed), or can be automatically determined. In the second case, several criteria can be used for optimal group number determination, such as the one described in [22].

In order to find the M most similar to a test vector $\mathbf{z}_{test}$ vector groups, we calculate the Euclidean distances between $\mathbf{z}_{test}$ and $\boldsymbol{\mu}_k$:

$$d_k = \|\mathbf{z}_{test} - \boldsymbol{\mu}_k\|^2. \tag{6}$$

After calculating $d_k,\ k = 1, ..., K$, the M most similar vector groups to the test vector $\mathbf{z}_{test}$ are those providing the M smallest distance values. M can either be assumed to be known (fixed), or can be automatically determined by following the procedure described in [15].

4. Extreme Learning Machine

Extreme Learning Machine (ELM) [18] is a fast algorithm for SLFNs training. In this section, we will provide an overview of ELM algorithm and discuss implementation issues appearing in our application setting. Let $\mathcal{X} = \{\mathbf{x}_i\}_1^{N_x}$ be a set of vectors, accompanied with the corresponding action class label set $\mathcal{C} = \{c_i\}_{i=1}^{N_x}\ c_i \in \mathcal{A}$. The network's target vectors corresponding to each vector $\mathbf{x}_i$, $\mathbf{t}_i = [t_{i1}, ..., t_{iC}]^T$, are set to $t_{ik} = 1$ for vectors belonging to action class k, i.e., when $c_i = k$, and $t_{ik} = -1$ otherwise.

In ELM, the network's input weights $\mathbf{W}_{in}$ are randomly chosen, while the output weights $\mathbf{W}_{out}$ are analytically calculated. Let us assume that the network's hidden layer consists of Q neurons and that $\mathbf{b} \in \mathbb{R}^Q$ is a vector containing the hidden layer neurons bias values, which are randomly chosen as well. Many activation functions $G()$ can be used for the hidden layer neurons' output calculation, such as sigmoid, sine, Gaussian and hard-limiting function. In our experi-

ments we have used the sigmoid function. That is, in our case $G(\mathbf{w}_j, b_j, \mathbf{x}_i) = \frac{1}{1+\exp^{-(\mathbf{w}_j^T \mathbf{x}_i + b_j)}}$, where $\mathbf{w}_j$ denotes the j-th column of $\mathbf{W}_{in}$. By storing the hidden layer neurons outputs in a matrix $\mathbf{G}$, i.e.,:

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{w}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{w}_1, b_1, \mathbf{x}_{N_x}) \\ \cdots & \ddots & \cdots \\ G(\mathbf{w}_Q, b_Q, \mathbf{x}_1) & \cdots & G(\mathbf{w}_Q, b_Q, \mathbf{x}_{N_x}) \end{bmatrix}, \tag{7}$$

the network's output vector corresponding to the training vector $\mathbf{x}_i$ can be written as $\mathbf{o}_i = \mathbf{W}_{out}^T \mathbf{g}_i$, where $\mathbf{g}_i$ denotes the i-th column of $\mathbf{G}$. The network's outputs corresponding to the entire vector set $\mathcal{X}$ can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{G}$. Finally, by assuming that the network's predicted outputs $\mathbf{O}$ are equal to the network's desired outputs $\mathbf{T}$, $\mathbf{W}_{out}$ can be analytically calculated by $\mathbf{W}_{out} = \mathbf{G}^\dagger \mathbf{T}^T$, where $\mathbf{G}^\dagger = (\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{G}$. However, the assumption of zero training error may decrease the generalization performance of the ELM network in the cases where the training set contains outliers. In order to increase the generalization performance of the ELM network, Huang et. al. [23] have recently proposed an optimization based regularized ELM algorithm formulated as follows:

$$\textbf{Minimize: } L_P = \frac{1}{2}\|\mathbf{W}_{out}\|^2 + \Lambda \frac{1}{2} \sum_{i=1}^{N_x} \|\boldsymbol{\xi}_i\|^2$$

$$\textbf{Subject to: } \mathbf{g}_i^T \mathbf{W}_{out} = \mathbf{o}_i^T - \boldsymbol{\xi}_i^T, \;\; i = 1, ..., N_x,$$

where $\boldsymbol{\xi}_i \in \mathbb{R}^C$ is a training error vector corresponding to training sample $\mathbf{x}_i$ and Λ is a parameter denoting the importance of the training error in the optimization problem. By adopting the above described optimization scheme, $\mathbf{W}_{out}$ can be calculated by:

$$\mathbf{W}_{out} = \left(\frac{1}{\Lambda}\mathbf{I} + \mathbf{G}\mathbf{G}^T\right)^{-1} \mathbf{G}\mathbf{T}^T. \tag{8}$$

After $\mathbf{W}_{out}$ calculation, a test vector $\mathbf{x}_{test}$ can be introduced to the ELM network and be classified to the class corresponding to the highest network's output, i.e.:

$$c_{test} = arg \max_j o_{test,j},\ j = 1, ..., C. \tag{9}$$

As can be seen the ELM training procedure is fast, since it involves matrix multiplication and matrix inversion operations. Such operations can be efficiently calculated by existing optimized software [24, 25]. Furthermore, the network topology and the input weights $\mathbf{W}_{in}$ can be determined only once, since they do not involve any training procedure.

5. Dynamic classification scheme

In this section we present the proposed dynamic classification method. Let $\mathcal{U}$ be an action recognition database, containing N action videos accompanied by the corresponding action class labels c_i, $i = 1, ..., N$ belonging to C action classes

forming an action class set $\mathcal{A}$. These action videos are preprocessed, following the procedure described in Section 2, in order to produce N action vectors $\mathbf{s}_i \in \mathbb{R}^D,\ i = 1, ..., N$.

Most classification schemes would employ all the available labeled action vectors $\mathbf{s}_i,\ i = 1, ..., N$ and the corresponding action class labels c_i in order to calculate a static classification model, that would be used in order to classify any unknown (test) action vector. In our case, the set of action vectors $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^{N}$ is clustered, by performing the procedure described in Section 3, in order to determine K action vector groups, represented by the corresponding mean group vectors $\boldsymbol{\mu}_k \in \mathbb{R}^D,\ k = 1, ..., K$.

Let a test action video be represented by an action vector $\mathbf{s}_{test} \in \mathbb{R}^D$. $\mathbf{s}_{test}$ is compared with all the K mean group vectors $\boldsymbol{\mu}_k$ in order to determine the M closest to $\mathbf{s}_{test}$ groups. The action vectors belonging to these M groups form the algorithm's first level training set $\mathcal{S}_1 = \{\mathbf{s}_{i,1}\}_{i=1}^{N_1}$. Here we have introduced a second index denoting the levels of the proposed dynamic classification scheme. Action class labels corresponding to the action vectors forming $\mathcal{S}_1$ are employed in order to form the first level action class label set $\mathcal{C}_1 = \{c_{i,1}\}_{i=1}^{N_1},\ c_{i,1} \in \mathcal{A}_1$. Obviously, $\mathcal{A}_1 \subseteq \mathcal{A}$, since only the labeled action vectors belonging to the action classes that are most similar to the actual $\mathbf{s}_{test}$ action class are included in $\mathcal{S}_1$. Now,

we can formulate an alternative classification problem. Instead of employing the entire action vector set $\mathcal{S}$ and train a universal classifier, we can use the action vector set $\mathcal{S}_1$ in order to train a $\mathbf{s}_{test}$ - specific classifier. That is, we train an SLFN by using $\mathcal{S}_1$ and $\mathcal{C}_1$ following the procedure described in Section 4. Subsequently, we introduce $\mathbf{s}_{test}$ to the trained network and we obtain its response $\mathbf{o}_{test}$. In this stage, we can classify $\mathbf{s}_{test}$ to the action class that provides the maximal network output, i.e.:

$$c_{test} = \underset{j}{argmax}\ o_{test,j}. \tag{10}$$

However, we choose to perform the dynamic classification procedure in multiple levels L. For this reason, we introduce $\mathcal{S}_1$ to the trained network and we obtain its responses $\mathcal{O}_1 = \{\mathbf{o}_{i,1}\}_{i=1}^{N_1}$. By using $\mathcal{O}_1$, we can now reformulate the classification problem. In the general case, after obtaining the l-th level network outputs, $\mathcal{O}_l = \{\mathbf{o}_{i,l}\}_{i=1}^{N_l}$ and $\mathbf{o}_{test,l}$, the feature vectors forming $\mathcal{O}_l$ are grouped by following the procedure described in Section 3. $\mathbf{o}_{test,l}$ is, subsequently, compared with the corresponding mean group vectors $\boldsymbol{\mu}_{k,l}$ and the closest to $\mathbf{o}_{test,l}$ groups are used to form the $(l+1)$-th level training set $\mathcal{S}_{l+1}$. The $(l+1)$-th level network is, subsequently, trained by using $\mathcal{S}_{l+1}$ and the corresponding action class label set $\mathcal{C}_{l+1} = \{c_{l+1,i}\}_{i=1}^{N_{l+1}},\ c_{l+1,i} \in \mathcal{A}_{l+1}$. Obviously the number of action classes forming the classification problem of every level of the proposed dynamic clas-

sification scheme is a subset of the number of classes forming the previous level classification problem, i.e., $\mathcal{A}_{l+1} \subseteq \mathcal{A}_l$.

The above described iterative procedure is performed multiple times, until the vectors forming the network training set belong to one action class only. That is, the maximal number of classification levels L depends on the test action vector $\mathbf{s}_{test}$. In the cases where the action class that $\mathbf{s}_{test}$ belongs to, is well distinguished from all the other action classes forming $\mathcal{A}$, only one classification level will be performed. In the cases of overlapping action classes, multiple classification levels will be performed in order to obtain the final classification result. Since at each level of the dynamic classification procedure the network training set is a subset of the previous level network training set, i.e., $\mathcal{S}_l \subset \mathcal{S}_{l-1}$, and the number of available labeled action vectors is finite, the proposed iterative procedure will converge in a finite number of iterations. In the, extreme, case of highly overlapping action classes, the iterative procedure will end when the network training set consists of only one labeled vector.

Consider the example illustrated in Figure 3. In this Figure, we illustrate the 2-dimensional feature space resulted by applying Principal Component Analysis (PCA) [11] on the dyneme based action video representation, in the Weizemann action recognition database [26], which will be used in the first set of the experi-

ments presented in the following section.

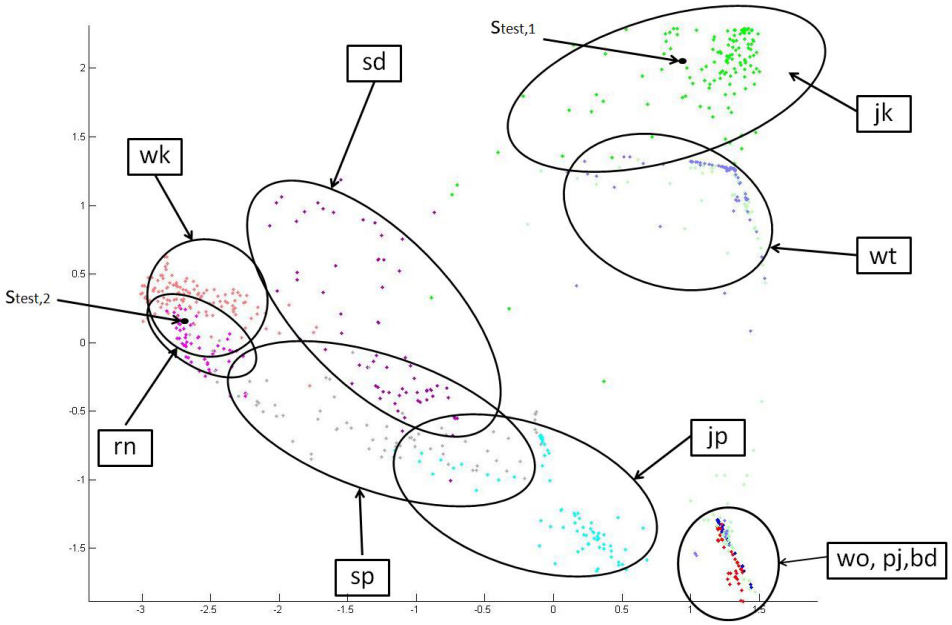


Figure 3: 2*D space resulted by applying PCA on the dyneme based action representation in the Weizemann action recognition database.*

As can be seen in Figure 3, some action classes, such as 'jumping jack' (jk), may be well distinguished from all the other action classes. However, action classes, usually, are confused with each other. Similar action classes, such as 'walk' (wk) and 'run' (rn), or 'skip' (sp) and 'jump in place' (jp) contain a high number of common human body poses and, thus, variations in action execution style and human body size may result to similar action representations. Assume that a test action video, represented by the corresponding action vector $\mathbf{s}_{test,1}$, be-

longs to the action class 'jumping jack' (jk). In this case, it is expected that $\mathbf{s}_{test,1}$ will be directly classified to the correct action class, since 'jumping jack' is well distinguished from all the other action classes. However, in the case of a test action video belonging to the action class 'run' (rn), represented by the action vector $\mathbf{s}_{test,2}$, its classification procedure is not obvious, since action class 'run' is confused with action class 'walk'. Thus, in this case, the classification procedure will probably involve multiple classification levels.

In Figure 3, it can also be seen that action classes are not linearly separable. For example, consider the case of action classes 'walk' and 'run', as highlighted in Figure 4. Clearly, these two action classes share the same feature space and are not linearly separable. Thus, the use of linear models for action class discrimination is not an appropriate choice. This can be seen in Figure 5, where we illustrate the separating hyperplanes (lines) resulted by applying LDA (Figure 5a) and ELM (Figure 5b) based action vectors classification, respectively. It can be seen that by applying the ELM based action vector classification, action classes are better discriminated. This is reasonable, since, as it was previously discussed, the use of ELM can better capture the non linear nature of the action classes.

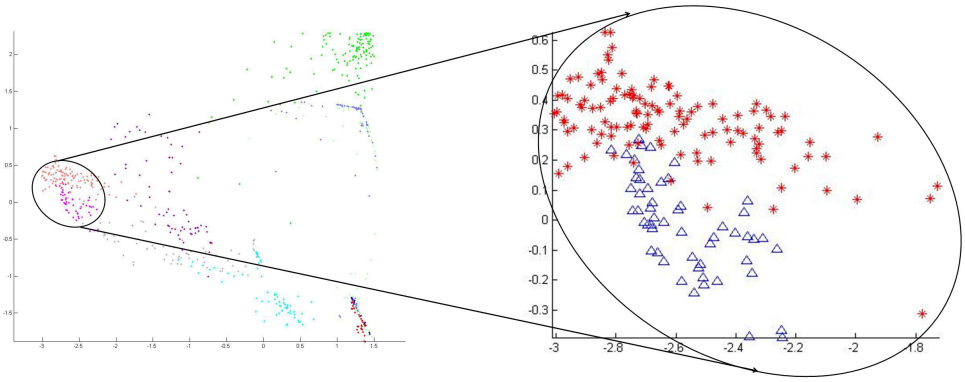
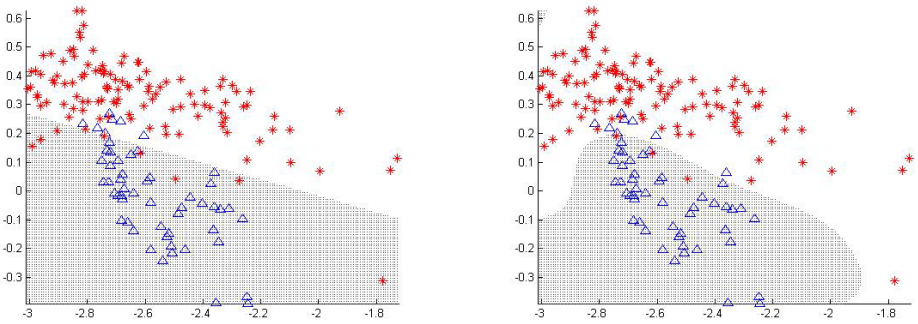


Figure 4: *Action vectors belonging to action classes 'walk' and 'run' in the* 2*D PCA space.*



(a) (b)

Figure 5: *Separating planes (lines) resulted by: a) applying LDA followed by nearest class centroid and b) training a single hidden layer neural network using the ELM algorithm.*

6. Experimental results

In this Section we present experiments conducted in order to evaluate the proposed dynamic classification method. We conducted experiments on the Weizemann [26] and the i3DPost [27] action recognition databases containing daily ac-

tions, as well as on a new action recognition database aiming at recognition of actions appearing in meal intakes [28]. We provide a comprehensive description of these databases in Subsections 6.1, 6.3 and 6.5, respectively. In each level of the proposed dynamic classification scheme, we grouped the labeled vectors in $K = [10, 20, 50]$ groups. The optimal number of closest to the test vector groups has been experimentally determined by using different values of $M = \frac{K}{k},\ k = 1, ..., 10$. Regarding the optimal number of dynemes D, the number of network hidden layer neurons Q and the parameter value Λ of the ELM algorithm, they have been determined by performing the leave-one-out cross-validation procedure. Specifically, we have performed the cross-validation procedure using values of D equal to $10k,\ k = 1, ..., 20$, $Q = [100, 200, 500, 1000]$ and values of Λ equal to $10^{\lambda},\ \lambda = -5, ..., 5$. In order to assess the ability of the proposed classification scheme to generalize on data that it was not trained on, we performed the leave-one-person-out cross-validation procedure (LOPOCV). That is, we used the action videos depicting all but one person in the database as labeled data and the action videos depicting the remaining one as test data, in order to perform one iteration (fold) of the cross validation procedure. Multiple folds, equal to the number of persons appearing in the database, have been performed in order to complete an experiment.

6.1. Weizemann database

The Weizemann action recognition database [26] contains 90 low-resolution, 144×180 pixel, image sequences depicting nine persons (five males and four females) performing ten daily actions each. The actions appearing in the database are: 'walk' (wk), 'run' (rn), 'jump in place on two legs' (pj), 'jump forward on two legs' (jp), 'jumping-jack' (jk), 'gallop sideways' (sd), 'skip' (sp), 'wave one hand' (wo), 'wave two hands' (wt) and 'bend' (bd). Binary image sequences denoting the human body regions are included in the database. Example video frames and binary skin-colored regions are illustrated in Figure 2.

Since most of these image sequences depict multiple action instances, e.g. multiple walking steps, we automatically produced binary action videos by using the binary image sequences and a sliding window consisting of 16 video frames, moving in steps of 4 video frames, resulting to the creation of 952 action videos. Figure 6 illustrates the sliding window technique for automatic binary action video creation. The resulted binary action videos have been preprocessed following the procedure described in Section 2

6.2. Experiments on the Weizemann database

In our first set of experiments we have conducted the LOPOCV procedure on the Weizemann action recognition database using the resulted binary action

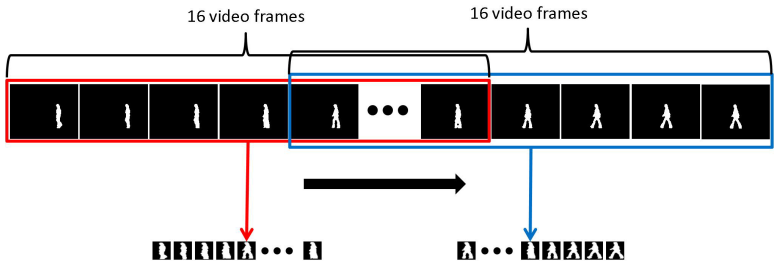


Figure 6: *Sliding window technique for automatic action video creation.*

videos. In Figure 7 we illustrate the action classification rates obtained by using different values of M. As can be seen, by using smaller values of M the classification rate increases. This is reasonable since for smaller values of M the classification procedure involves only the labeled data that are more similar to the test ones.

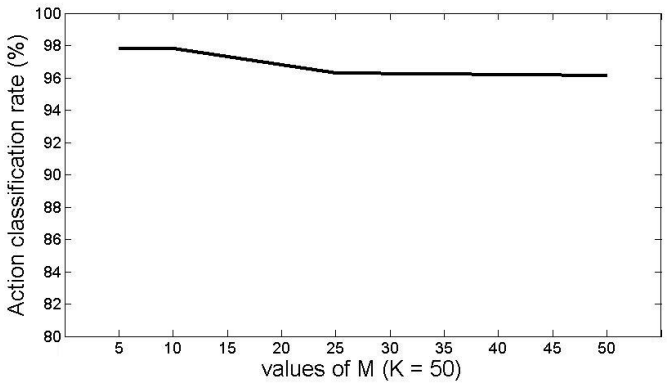


Figure 7: *Action classificaction rates as a function of* M.

The confusion matrix corresponding to the optimal parameters is illustrated in Figure 8a. As can be seen, high classification rates have been obtained for all the

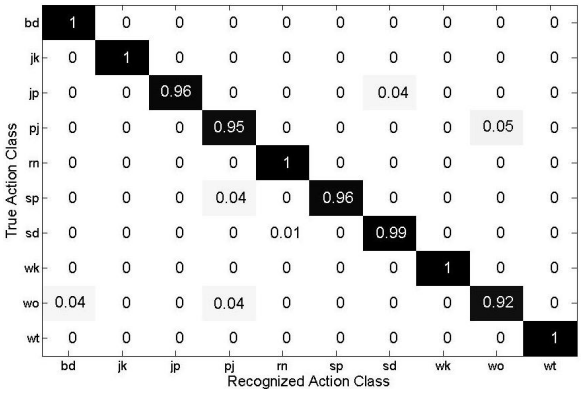
action classes. The class which was found to be the most difficult for classification is action class 'wave one hand', which is confused with action classes 'bend' and 'jump in place on two legs'. However, even for this case a high classification rate, equal to 92%, has been obtained.

In order to directly compare the performance of the proposed classification method with other ones, we have conducted experiments by performing the LOPOCV procedure on the Weizemann action recognition database using both static and dynamic classification strategies. That is, we performed the LOPOCV procedure by employing the static classification strategy and performing LDA based action vector projection followed by nearest class centroid classification, resulting to an action classification rate equal to 95.92%. By following the static classification strategy and performing ELM based action vector classification, an action classification rate equal to 96.15% has been obtained.

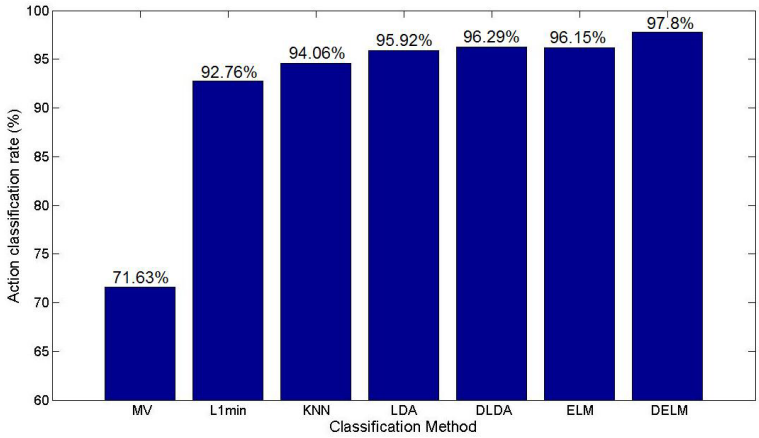
Subsequently, we have conducted experiments employing the dynamic classification strategy. KNN action vectors classification, using $K = 3$ nearest neighbors, resulted to an action classification rate equal to 94.06%. Action classification based on L1-minimization followed by smallest residual error action vector classification, as proposed by Wright et. al. [13], resulted to an action classification rate equal to 92.76%. By following the dynamic classification method proposed

by Kyperountas et. al. [15], which employs LDA based data projection, an action classification rate equal to 96.29% has been obtained. Finally, by performing one level of the proposed dynamic classification method and classifying each test action vector by applying majority voting on the action class labels of the labeled action vectors forming the M, out of K, closest to the test action vector groups, an action classification rate equal to 71.63% has been obtained. This procedure can be used as a reference for the performance of the proposed dynamic classification scheme, since, intuitively, the determination of labeled action vectors similar to the test one should lead to correct classification results. The action classification rates obtained in all these experiments are summarized in Figure 8b.

As can be seen in Figure 8, the adoption of a dynamic classification strategy leads to an increase of the action classification rates. In both the LDA and ELM cases, the dynamic classification approach provides higher classification rates. Furthermore, it can be seen that the proposed classification scheme is efficient, since a simple majority voting on the action labels of the labeled action vectors that form the M most similar to the test action vectors groups, results to a, relatively, high action classification rate. Finally, it can be seen that the proposed dynamic classification scheme outperforms all the other competing methods appearing in Figure 8, since it combines both efficient search of the most appropriate



(a)



(b)

Figure 8: *a) Confusion matrix on the Weizemann database obtained by applying the proposed dynamic action classification method and b) Comparison results on the Weizemann action recognition database.*

training set and approximation of non-linear discrimination functions.

In order to compare the performance of the proposed action classification method with that of other methods proposed in the literature, we have followed

the procedure proposed in [29]. That is, the image sequences of the Weizemann database have been classified to action classes by performing majority voting on the action classification results provided by the algorithm for the corresponding action videos, resulting to an action classification rate equal to 98.9%. All but one action sequences have been correctly classified. The only sequence that was misclassified belongs to action class 'skip' and classified to action class 'jump forward on two legs'. Since some of the methods proposed in the literature providing state of the art performance are evaluated by using an earlier version of the database containing nine action classes, i.e., not containing action class 'skip', we have also tested the proposed dynamic action classification method by using this earlier version. Comparison results with other action recognition methods are illustrated in Table 1.

6.3. AIIA-MOBISERV database

Despite the fact that most applications, including action recognition functionality, consider daily action types, such as walk, run, etc., there are applications requiring different type of actions. For example, monitoring the status of the elderly people in the early stages of dementia, while still living independently, to prevent dehydration is an important task. In the framework of the EU R&D project MOBISERV, we created an eating and drinking action recognition database, which

Table 1: Comparison results on the Weizmann action recognition database.

Method	9 actions	10 actions
Yaffet & Wolf [30]	100%	–
Wang & Mori [31]	100%	–
Guha & Ward [32]	–	98.9%
Gorelic et al. [29]	–	97.8%
Riemenchneider et al. [33]	–	96.7%
Gkalelis et.al. [34]	–	96%
Ali & Shah [35]	–	95.7%
Junejo et al. [36]	95.3%	–
Thurau & Hlavac [37]	–	94.4%
Zhang et al. [38]	–	92.8%
Niebles et al. [39]	–	90%
Proposed method	**100%**	**98.9%**

is publicly available in [28]. Twelve persons (six females and six males) were captured by a camera placed at a distance of 2 meters in front of them, during a meal. Four meals have been recorded, each for a different day for all the twelve persons. The actions appearing in the database are: 'eat', 'drink' and 'apraxia'. Action class 'eat' contains the cases where the person eats using a spoon, a cut-

lery, a fork, or takes a bite using one or two hands. Action class 'drink' contains the cases where the person drinks using a cup, a glass, or a straw. Finally, action class 'apraxia' contains the cases where the person is slicing his/her food or he/she is chewing it and the cases where the person rests.

We have manually temporally segmented the videos depicting all the persons during two meals. This procedure resulted to the creation of 1288 action videos. A color based image segmentation technique has been applied to the video frames of these action videos in order to produce binary images depicting the skin regions of the depicted person's body. Specifically, each video frame has been converted to the HSV color space and the image pixels having HS values in pre-specified thresholds, corresponding to skin-like color values, have been determined to be foreground pixels, while the rest pixels have been assumed to belong to the background. Morphological operations (closing) have been, subsequently, performed in order to obtain the final binary action video frames. This resulted to the creation of binary action videos denoting the person's head and hands, which have been preprocessed following the procedure described in Section 2. Example video frames and binary skin-colored regions are illustrated in Figure 2.

6.4. Experiments on the AIIA-MOBISERV database

In our second set of experiments we have performed the LOPOCV procedure on the binary action videos of the AIIA-MOBISERV database. An action classification rate equal to 93.4% has been obtained by applying the proposed dynamic action classification method. The confusion matrix of this experiment is illustrated in Figure 9a. Comparison results with other dynamic, as well as static, action classification schemes are illustrated in Figure 9b. As can be seen, by applying the majority voting classification scheme, an action classification rate equal to 65.11% has been obtained. Action classification based on L1-minimization followed by smallest residual error action vector classification, resulted to an action classification rate equal to 90.3%. KNN ($K = 3$) action vector classification resulted to an action classification rate equal to 89.91%. Static LDA and ELM based action classification schemes, provided action classification rates equal to 89.94% and 89.73%, respectively. Finally, LDA-based dynamic action vector classification resulted to an action classification rate equal to 92.53%. As can be seen, the dynamic action classification approach outperforms the static one in both the LDA and ELM based classification schemes. Furthermore, it can be seen that the ELM based dynamic action classification scheme outperforms all the methods presented in this Figure.

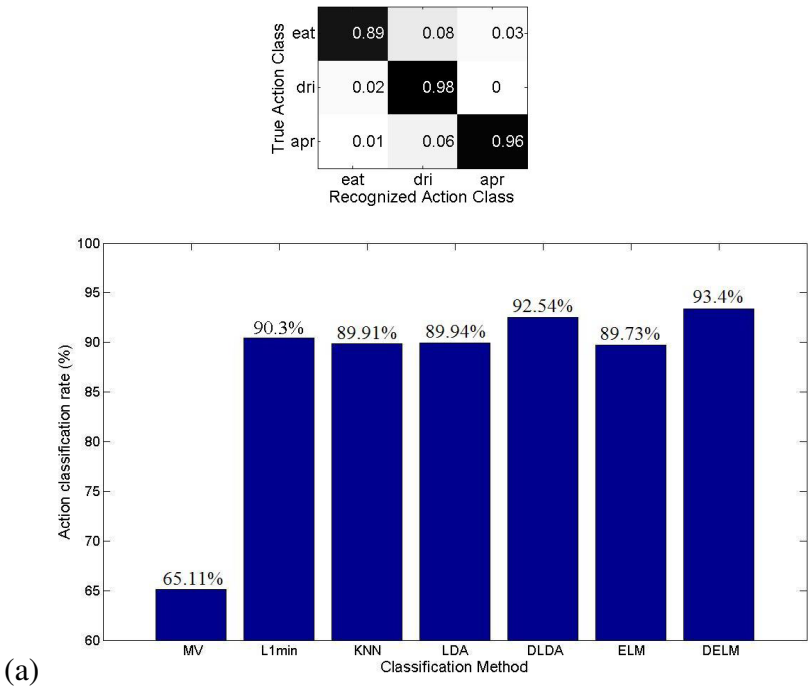


Figure 9: *a) Confusion matrix on the AIIA-MOBISERV database obtained by applying the proposed dynamic action classification method and b) Comparison results on the AIIA-MOBISERV action recognition database.*

6.5. i3DPost database

The i3DPost multi-view database [27] contains 512 high resolution (1080×1920 pixels) image sequences depicting eight persons (six males and two females) performing eight actions. The database camera setup consists of eight cameras, providing a 360° coverage of the scene. The actions appearing in the database are: 'walk' (wk), 'run' (rn), 'jump in place' (jp), 'jump forward' (jf), 'bend' (bd), 'fall

down' (fl), 'sit on a chair' (st) and 'wave one hand' (wo). Since most of the image sequences depict multiple action instances, e.g. multiple walking steps, we have manually temporally segmented them in order to produce videos depicting one action instance each. A color based image segmentation technique, discarding the blue color in the HSV color space, has been applied to the video frames of these action videos in order to produce binary action videos denoting the human body. Morphological operations (closing) have been, subsequently, performed in order to obtain the final binary action video frames.

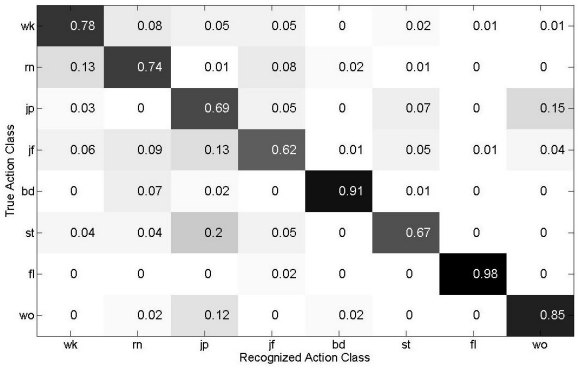
6.6. Experiments on the i3DPost database

In our third set of experiments we have performed the LOPOCV procedure on the binary action videos of the i3DPost database. In each fold of the LOPOCV procedure, we have used the action videos depicting seven of the persons performing an action instance from all the available cameras as labeled data. Each action video depicting the test person has been classified to one of the eight action classes independently, in order to form a single-view view-invariant action classification problem. We should note that, we expected the above described procedure to result to a difficult classification problem due to the well known view angle effect [40]. By applying the proposed dynamic action classification method, an action classification rate equal to 77.97% has been obtained. The confusion matrix of this

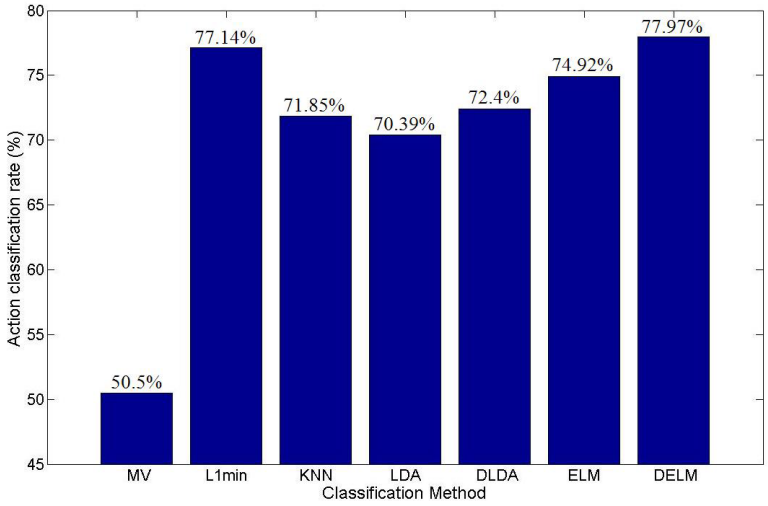
experiment is illustrated in Figure 10a. Comparison results with other dynamic, as well as static, action classification schemes are illustrated in Figure 9b. As can be seen, by applying the majority voting classification scheme, an action classification rate equal to 50.5% has been obtained. Action classification based on L1-minimization followed by smallest residual error action vector classification, resulted to an action classification rate equal to 77.14%. KNN ($K = 3$) action vector classification resulted to an action classification rate equal to 71.85%. Static LDA and ELM based action classification schemes, provided action classification rates equal to 70.39%. Finally, LDA-based dynamic action vector classification resulted to an action classification rate equal to 72.4%. As can be seen, the dynamic action classification approach outperforms the static one in both the LDA and ELM based classification schemes. Furthermore, it can be seen that the ELM based dynamic action classification scheme outperforms all the methods presented in this Figure.

7. Conclusions

In this paper, we proposed a novel dynamic action classification method based on an iterative procedure determining test action instance specific classification problems in multiple levels. Action instances are represented by vectors denoting



(a)



(b)

Figure 10: *a) Video frames depicting one person of the i3DPost database walking from different viewing angles and b) Comparison results on the i3DPost action recognition database.*

the fuzzy similarity of the corresponding human body poses with representative human body poses, the dynemes. At each classification level, the most similar to the test action instance labeled vectors are employed in order to train a single

hidden layer feedforward network using the ELM algorithm. A new feature space is, subsequently, obtained by the trained network's outputs. By exploiting the properties of the adopted network topology and the fast training procedure of the ELM algorithm, the proposed classification method is fast and efficient. Experiments on publicly available action recognition databases indicate the superiority of the dynamic classification strategy, compared to the static one, as well as the effectiveness of the proposed dynamic classification method.

Acknowledgment

The research leading to these results has received funding from the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

References

[1] L. Weilun, H. Jungong, P. With, Flexible Human Behavior Analysis Framework for Video Surveillance Applications, International Journal of Digital Multimedia Broadcasting 2010 (2010) 9, ISSN 1687-7578.

[2] P. Barr, J. Noble, R. Biddle, Video game values: Human-computer interaction and games, Interacting with Computers 19 (2) (2007) 180–195.

[3] T. Hollerer, S. Feiner, D. Hallaway, B. Bell, M. Lanzagorta, D. Brown, S. Julier, Y. Baillot, L. Rosenblum, User interface management techniques for collaborative mobile augmented reality, Computers and Graphics 25 (5) (2001) 799–810.

[4] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, W. Nunziati, Semantic annotation of soccer videos: automatic highlights identification, Computer Vision and Image Understanding 92 (2-3) (2003) 285–305.

[5] S. Ali, M. Shah, Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2) (2010) 288–303.

[6] Y. Wang, G. Mori, Hidden Part Models for Human Action Recognition: Probabilistic versus Max Margin, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1310–1323.

[7] M. Giese, T. Poggio, Neural mechanisms for the recognition of biological movements, Nature Reviews Neuroscience 4 (3) (2003) 179–192.

[8] A. Iosifidis, A. Tefas, N. Nikolaidis, I. Pitas, Multi-view human movement recognition based on Fuzzy distances and Linear Discriminant Analysis, Elsevier Computer Vision and Image Understanding 116 (2012) 347–360.

[9] A. Iosifidis, A. Tefas, I. Pitas, View-Invariant Action Recognition Based on Artificial

Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 23 (3) (2012) 412–425.

[10] S. Haykin, Neural networks and learning machines, Upper Saddle River, New Jersey .

[11] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed, Wiley-Interscience, 2000.

[12] A. Iosifidis, A. Tefas, I. Pitas, Person specific activity recognition using fuzzy learning and discriminant analysis, European Signal Processing Conference (EUSIPCO) (2011) 1974–1978.

[13] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[14] H. Tang, M. Lyu, I. King, Face recognition committee machines: dynamic vs. static structures, Proceedings of 12th International Conference on Image Analysis and Processing (2003) 121–126.

[15] M. Kyperountas, A. Tefas, I. Pitas, Dynamic training using multistage clustering for face recognition, Pattern Recognition 41 (3) (2008) 894–905.

[16] J. Lu, K. Plataniotis, A. Venetsanopoulos, Face recognition using LDA-based algorithms, IEEE Transactions on Neural Networks 14 (1) (2003) 195–200.

[17] J. Lu, K. Plataniotis, A.N.Venetsanopoulos, Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition, Pattern Recognition Letters 26 (2) (2005) 181–191.

[18] G. Huang, Q. Zhu, C. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, IEEE International Joint Conference on Neural Networks 2 (2004) 985–990.

[19] N. Gkalelis, A. Tefas, I. Pitas, Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1511–1521.

[20] M. Seki, T. Wada, H. Fujiwara, K. Sumi, Background subtraction based on the cooccurrence of image variations, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2 (2003) 65–72.

[21] V. Vezhnevets, V. Sazonov, A. Andreeva, A survey on pixel-based skin color detection techniques, Proc. Graphicon 3.

[22] B. Frey, D. Dueck, Clustering by passing messages between data points, science 315 (5814) (2007) 972.

[23] G. Huang, H. Zhou, X. Ding, R. Zhang, Extreme Learning Machine for Regression

and Multiclass Classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (2) (2012) 513–529.

[24] G. Bradski, A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, 2008.

[25] http://icl.cs.utk.edu/lapack-for-windows/ .

[26] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, Proceedings of IEEE Conference on Computer Vision 2 (2005) 1395–1402.

[27] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, I. Pitas, The i3DPost multi-view and 3D human action/interaction database, 6th Conference on Visual Media Production (2009) 159–168.

[28] http://poseidon.csd.auth.gr/MOBISERV-AIIA/index.html .

[29] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as Space-Time Shapes, Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[30] L. Yeffet, L. Wolf, Local trinary patterns for human action recognition, International Conference on Computer Vision (2009) 492–497.

[31] Y. Wang, G. Mori, Human action recognition by semilatent topic models, IEEE

Transactions on Pattern Analysis and Machine Intelligence 31 (10) (2009) 1762–1774.

[32] T. Guha, R. Ward, Learning Sparse Representations for Human Action Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence .

[33] H. Riemenschneider, M. Donoser, H. Bischof, Bag of optical flow volumes for image sequence recognition, British Machine Vision Conference (BMVC) .

[34] N. Gkalelis, A. Tefas, I. Pitas, Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1511–1521, ISSN 1051-8215.

[35] S. Ali., M. Shah, Human action recognition in videos using kinematic features and multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2) (2010) 288–303.

[36] I. Junejo, E. Dexter, I. Laptev, P. Pérez, View-independent action recognition from temporal self-similarities, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (1) (2011) 172–185.

[37] C. Thurau, V. Hlavác, Pose primitive based human action recognition in videos or

still images, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (2008) 1–8.

[38] Z. Zhang, Y. Hu, S. Chan, L. Chia, Motion context: A new representation for human action recognition, European Conference on Computer Vision ECCV (2008) 817–829.

[39] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, International Journal of Computer Vision 79 (3) (2008) 299–318.

[40] S. Yu, D. Tan, T. Tan, Modeling the effect of view angle variation on appearance-based gait recognition, Proceedings Asian Conf. Computer Vision 1 (2006) 807–816.